{EPITECH}

GOMOKU_

< SMART BOTS FOR A SIMPLE GAME />

GOMOKU

Binary name: pbrain-gomoku-ai
Language: Anything as long as it works on ”the dump”
Compilation: When necessary, via Makefile, including re, clean and fclean rules

- ✓ The totality of your source files, except all useless files (binary, temp files, obj files,...), must be included in your delivery.
- ✓ All the bonus files should be in a directory named bonus.

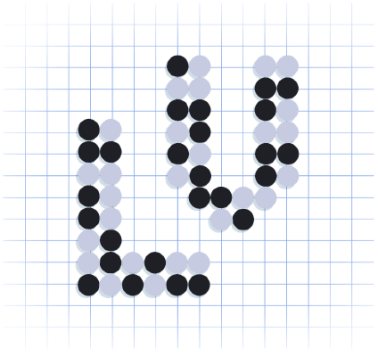
The goal of this project is to implement a *Gomoku Narabe* game bot (also called *Wuzi Qi*, *Slope*, *Darpion* or *Five in a Row*), focusing on the performance of its artificial intelligence.

Your bot must be compliant with the communication protocol (Epitech Mirror), at least the **Mandatory commands** part.

To run your bot you will need to use a game manager, here we will use liskvork which is entirely Free and Open Source, and has a wide variety of supported platforms/architecture.

liskvork is freely available on the internet as binaries or directly as source code (Epitech Mirror).

To get you started some very basic AI templates are available here for C++, Zig, Rust, Python...



Feel free to implement an algorithm of your choice for your bot (Min-max, Monte-Carlo, Machine Learning or other). You will be evaluated on the efficiency of your bot, and on this criteria alone.

You may need to develop a rules management algorithm; do not hesitate to enrich your board representation and your data structures to optimize this algorithm!

There are some technical constraints you must comply with:

- ✓ whatever development language you choose, your program must compile **on Linux** using your makefile (if compilation is needed)
- ✓ **only standard libraries are allowed. No tensorflow or scikit-learn here**

Game Rules

This is a 2-player game that is played on a 20x20 game board (called **goban**). We use the freestyle ruleset for this project. Every player plays a stone on their turn, and the game ends as soon as one has 5 stones in a row (vertically, horizontally or diagonaly) - and thus wins.

Evaluation

Your bot will be evaluated based on its results in actual game playing, via a 2-step process:

- ✓ **play to win**
  In order to validate the module, your program is intended to understand some winning situations and play the right move. We will send you pre-filled board and your program must make the right move, for example if there is a winning move your program must play it.
- ✓ **outsmart local bots**
  Programs are matched against bots of low to medium levels. The aim is to maximize your number of victories.

Brawl

This project also includes a tournament that is completely **optional** and **opt-in** named Brawl.
It does not impact your grade if you play or not in it.

Access URL: `https://gomoku.epitest.eu`

How to opt-in

To opt-in create a file named ".brawl" at the root of your repository (it is a hidden file, the dot at the start is on purpose).
In it you can put the name of your AI, which will be displayed on the tournament's website.

The name is a maximum 16 characters long, a minimum of 5 characters, and only contain alpha-numeric characters, spaces, dashes and underscores.
Here is a validation regex for it: `^[a-zA-Z _-]{5,16}$`

If you need to compile make sure that your make's clean rule does not delete your binary!
If it does your AI will be marked as not building in the tournament.

Features

There are quite a few features inside the tournament website:
- Global national leaderboard
- Elo system (standard bayesian Elo system)
- Complete game list
- Elo over time analysis per brain with game stats (such as the win ratio)
- Replay system (you can watch any match played on the tournament, or locally)
- The solution to every single problem in the world (ok, maybe not this one :p)

Did you know that Brawl has a public API?

Technical constraints

During the whole process, the rules are as follows:
- **5 seconds** maximum per move,
- **70 MB** of memory per bot,
- a forbidden move automatically leads to defeat,
- the use of a forbidden library leads to elimination.

The rules are the same for the tournament and the automated tests!

v 6